Tutorial 11: $O()$ and $\Theta()$ Notation

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

Informal definition

We have seen the formal definitions of
(a) $f(x) = O(g(x))$ and (b) $f(x) = \Theta(g(x))$.

Informally

$f(x) = O(g(x))$ means that
$f(x)$ grows no faster than $g(x)$

and

$f(n) = \Theta(g(x))$ means that
$f(x)$ grows like $g(x)$.

We will now see more about using this notation.

Formal definition of $O()$

Recall the definition:

Function $f(x) = O(g(x))$ read: $f(x)$ is O of $g(x)$):

If (i) There is some positive $x_0 \in R$
(ii) There is some positive $c \in R$
such that
$\forall x \geq x_0 \quad f(x) \leq cg(x).$

Some examples

$4x^2$
$8x^2 + 2x - 3$
$x^2/5 + \sqrt{x} - 10 \log x$
$x(x - 3)$
are all $O(x^2)$.

Statements like these can often be proven using simple tools;

It is NOT usually necessary to prove $f(x) = O(g(x))$ from scratch!

Observation 1

If $\exists x_0$ and $\exists c > 0$ such that

$$\forall x > x_0, f(x) \leq c$$

Then $f(x) = O(1)$

This comes directly from definition of $O(1)$.

Examples:

$$\begin{aligned} \sin(x) &= O(1) \\ 2 + \frac{1}{x} &= O(1) \end{aligned}$$

Observation 2

"Constants don't matter"

If $f(x) = O(g(x))$ then

$\forall c > 0, f(x) = O(cg(x))$

Proof: $f(x) = O(g(x))$ means that $\exists x_0, c'$ such that

$$\forall x > x_0, f(x) \leq c' g(x).$$

But then,

$$\forall x > x_0, f(x) \leq \frac{c'}{c} cg(x),$$

so $f(x) = O(cg(x))$

Examples:

If $f(x) = O(\frac{1}{2}x^2)$ then $f(x) = O(x^2)$.
If f(x) = O(5) then $f(x) = O(1)$.

Observation 3

If $f_1(x) = O(g_1(x))$ and $f_2(x) = O(g_2(x))$ then

$$f_1(x) + f_2(x) = O(g_1(x) + g_2(x))$$

Proof: By definition, there exist x_1, c_1, x_2, c_2, such that

$$\begin{aligned} \forall x &\geq x_1 \quad f_1(x) \leq c_1 g_1(x). \\ \forall x &\geq x_2 \quad f_2(x) \leq c_2 g_2(x). \end{aligned}$$

Letting $c_3 = \max\{c_1, c_2\}$ and $x_3 = \max\{x_1, x_2\}$ gives

$$\forall x \geq x_3 \quad f_1(x) + f_2(x) \leq c_3(g_1(x) + g_2(x)).$$

Example:

$$\begin{aligned} f_1(x) &= x^2 + 3x & f_1(x) &= O(x^2). \\ f_2(x) &= 5x + 7 & f_2(x) &= O(x). \end{aligned}$$

Then $x^2 + 8x + 7 = f_1(x) + f_2(x) = O(x^2 + x) = O(x^2)$

The last equality is obtained by observation 2.

Observation 4

Suppose $f(x) = O(g(x) + h(x))$ and $h(x) = O(g(x))$. Then

$$f(x) = O(g(x))$$

This follows directly from the definition of $O(g(x))$.

Example:

Suppose $f(x) = O(x^2 + x)$. Since $x = O(x^2)$ we get

$$f(x) = O(x^2)$$

Observation 5

If $f_1(x) = O(g_1(x))$ and $f_2(x) = O(g_2(x))$ then

$$f_1(x) \cdot f_2(x) = O(g_1(x) \cdot g_2(x))$$

Proof: By definition, there exist x_1, c_1, x_2, c_2, such that

$$\begin{aligned} \forall x \geq x_1 \qquad & f_1(x) \leq c_1 g_1(x). \\ \forall x \geq x_2 \qquad & f_2(x) \leq c_2 g_2(x). \end{aligned}$$

Letting $c_3 = c1 \cdot c2$ and $x_3 = \max\{x_1, x_2\}$ gives

$$\forall x \geq x_3 \quad f_1(x) \cdot f_2(x) \leq c_3(g_1(x) \cdot g_2(x)).$$

Observation 5

Example 1:

$$f_1(x) = x^2 + 3x + 5 \qquad f_1(x) = O(x^2).$$
$$f_2(x) = x + 7 \qquad f_2(x) = O(x).$$

Then $(x^2 + 3x + 5)(x + 7) = O(x^2 \cdot x) = O(x^3)$

Example 2:

In class we learnt that $\sum_{i=0}^{n-1} r^i = \begin{cases} O(r^n) \text{ if } r > 1 \\ O(1) \text{ if } r < 1 \end{cases}$

This implies that

$$\begin{aligned} n\sum_{i=0}^{n-1} 3^i &= O(n3^n) \\ n\sum_{i=0}^{n-1} \left(\frac{1}{2}\right)^i &= O(n \cdot 1) = O(n) \end{aligned}$$

Pulling it all together

Suppose an algorithm runs in time

$$T(n) = nT(1) + 4n\sum_{i=0}^{n-1} 3^i.$$

Using our observations (skipping some steps), this can be rewritten as

$$\begin{aligned} T(n) &= O(n + n3^n) \\ &= O(n3^n) \end{aligned}$$

Pulling it all together

Suppose an algorithm runs in time

$$T(n) = nT(1) + 4n \sum_{i=0}^{\log_2 n - 1} 2^i.$$

First, recall that $2^{\log_2 n - 1} = \frac{1}{2}n$

$$\rightarrow \qquad \sum_{i=0}^{\log_2 n - 1} 2^i = O\left(\frac{1}{2}n\right) = O(n)$$

$$\rightarrow \qquad T(n) = O(nT(1) + 4n \cdot n) = O(n^2)$$

Formal definition of $\Theta()$

Two functions $f(x), g(x)$ have the same order of growth if $f(x) = O(g(x))$ and $g(x) = O(f(x))$.

In this case recall that we say

$$f(x) = \Theta(g(x))$$

which is the same as

$$g(x) = \Theta(f(x))$$

Observation 1'

If $\exists x_0$ and $\exists c_1, c_2 > 0$ such that

$$\forall x > x_0, \quad c_1 \leq f(x) \leq c_2$$

Then $f(x) = \Theta(1)$

Example:

$$2 + \frac{1}{x} = \Theta(1)$$

but

$$\sin(x) \neq \Theta(1)$$

The reason is that $\sin(x)$ is not bounded from below by a positive constant. It can approach arbitrarily close to 0.

Observation 2'

"Constants don't matter"

If $f(x) = \Theta(g(x))$ then

$$\forall c > 0, f(x) = \Theta(cg(x))$$

Examples:

If $f(x) = \Theta(\frac{1}{2}x^2)$ then $f(x) = \Theta(x^2)$.

If $f(x) = \Theta(5)$ then $f(x) = \Theta(1)$.

Observation 3'

If $f_1(x) = \Theta(g_1(x))$ and $f_2(x) = \Theta(g_2(x))$ then

$$f_1(x) + f_2(x) = \Theta(g_1(x) + g_2(x))$$

Example:

$$\begin{aligned} f_1(x) &= x^2 + 3x & \qquad f_1(x) &= \Theta(x^2). \\ f_2(x) &= 5x^2 + 7 & \qquad f_2(x) &= \Theta(x^2). \end{aligned}$$

Then

$$\begin{aligned} 6x^2 + 3x + 7 = f_1(x) + f_2(x) &= \Theta(x^2 + x^2) \\ &= \Theta(x^2) \end{aligned}$$

The last equality is obtained by observation 2'.

Observation 4'

Suppose $f(x) = \Theta(g(x) + h(x))$ and $h(x) = O(g(x))$. Then

$$f(x) = \Theta(g(x))$$

NOTE: It is only required that $h(x) = O(g(x))$.
We do NOT need $h(x) = \Theta(g(x))$!

Example:

Suppose $f(x) = \Theta(x^2 + x)$. Since $x = O(x^2)$ we get

$$f(x) = \Theta(x^2)$$

Observation 5'

If $f_1(x) = \Theta(g_1(x))$ and $f_2(x) = \Theta(g_2(x))$ then

$$f_1(x) \cdot f_2(x) = \Theta(g_1(x) \cdot g_2(x))$$

Example:

$$\begin{aligned} f_1(x) &= x^2 + 3x + 5 \qquad & f_1(x) &= \Theta(x^2) \\ f_2(x) &= x + 7 \qquad & f_2(x) &= \Theta(x) \end{aligned}$$

So

$$(x^2 + 3x + 5)(x + 7) = \Theta(x^2 \cdot x) = \Theta(x^3).$$

Pulling it all together

Suppose an algorithm runs in time

$$T(n) = nT(1) + 4n \sum_{i=0}^{\log_2 n - 1} 2^i.$$

First, recall that

$$\rightarrow \qquad \sum_{i=0}^{\log_2 n - 1} 2^i = 2^{\log_2 n} - 1 = n - 1 = \Theta(n)$$

$$\rightarrow \qquad T(n) = \Theta(nT(1) + 4n \cdot n) = \Theta(n^2)$$

Logs and $O()$ and $\Theta()$-notation

Recall that there is really no such function as $\log n$.
A log is determined by its base, e.g., $\log_2 n$ or $\log_{10} n$.
We also write $\ln n$ to denote $\log_e n$.

When using $O()$ and $\Theta()$ notation, people frequently don't write the base of the logarithm. For example, they might write $\Theta(n \log n)$ instead of $\Theta(n \log_3 n)$.

The reason this doesn't cause confusion is that, $\forall a, b > 0$

$$\log_a n = (\log_a b) \log_b n$$

e.g.,

$$\log_2 n = \log_2 4 \cdot \log_4 n = 2 \log_4 n$$

Logs and $O()$ and $\Theta()$-notation

Since, $\forall a, b > 0$

$$\log_a n = (\log_a b) \log_b n$$

we see that

$$\log_a n = \Theta(\log_b n).$$

In particular, this implies that it doesn't really matter which base of the log we write in the $O()$ or $\Theta()$ notation equation. So the base is unimportant.

For example, since $\log_a n = \Theta(\log_b n)$,

$$f(n) = \Theta(n \log_a n) \quad \textit{and} \quad f(n) = \Theta(n \log_b n)$$

are equivalent statements (since each implies the other).

So, some people would just write $f(n) = O(n \log n)$.